Evaluation of Table-based Access Control in IoT Data Distribution Method using Fog Computing

Masaki Yoshii\textsuperscript{1a),} Ryohei Banno\textsuperscript{2}, and Osamu Mizuno\textsuperscript{12}

\textsuperscript{1} Graduate School of Electric and Electronics Engineering, Kogakuin University
1-24-2, Nishi-Shinjuku, Shinjuku-ku, Tokyo 163-8677, Japan
\textsuperscript{2} Faculty of Informatics Department of Information and Communication Engineering, Kogakuin University, 2665-1, Nakanomachi, Hachioji-shi, Tokyo 192-0015, Japan

\textsuperscript{a) cm20053@ns.kogakuin.ac.jp}

Abstract: New services can use fog nodes to distribute Internet of Things (IoT) data. To distribute IoT data, we apply the publish/subscribe messaging model to a fog computing system. However, there is a possibility that the user’s private data are distributed without their permission. In this paper, we propose a Table-based Access Control List to protect unnecessary private data distribution. The evaluation results show that the bandwidth usage is reduced by about 40% and the queuing delay of a service fog node is reduced by about 60%.

Keywords: Fog Computing, IoT Data Distribution, Access Control

Classification: Network System

References


**1 Introduction**

Fog computing [1] deploys multiple fog nodes between IoT devices and the cloud to pre-process IoT data. New value-added services can be provided with IoT data distribution among multiple players via fog nodes. However, each player may deploy his fog node and IoT devices, the private data may distribute to others. Specifically, this is because the IoT data acquisition ID given by the Service Provider (SP) to the user has a correspondence relationship with multiple IoT devices [2].

Users apply Table-based ACL (Access Control List) to their UFN (User’s Fog Node) to protect unnecessary distribution of private data. The users can arbitrarily decide whether to “permit” or “deny” the transmission of IoT data toward destination SFN (Service Fog Node). It enables reliable protection against private data leakage.

In this paper, we evaluate the performance of the IoT data distribution method with Table-based ACL by simulation. This paper gives additional evaluations to our former paper [2]. This paper includes the queuing delay evaluations and average bandwidth usage of the SFN side by applying the proposed method.

**2 Tag ID-based publish/subscribe model**

We assume two players: the user and the SP. The users deploy UFNs and collect data from their IoT devices. SPs operate cloud servers and deploy SFNs. The users send IoT data via UFN to SFN based on their contractual relationships. However, to receive high-quality services, it may be beneficial for users to share their IoT data among SPs without any contractual relationships. In this paper, we describe a SP who has/doesn’t have a contractual relationship to the user as $SP_\alpha$ and $SP_\beta$, respectively. If $SP_\beta$ can use the user’s data, it would provide high-quality services. To realize the services, it is preferable that $SP_\beta$ get user’s ID from $SP_\alpha$ based on the contractual relationship between $SP_\alpha$ and $SP_\beta$. $SP_\beta$ uses this ID to acquire data from the user.

The Tag ID-based pub/sub (publish/subscribe) model [3] is a communication model for IoT data distribution. Here, Tag ID is an ID assigned by the SP to the topic, which is the ID for data acquisition defined by the user. The pub/sub model [4] is assumes to use for IoT data collection. When the number of topics is equal to the number of IoT devices, SPs need to store topics for the number of IoT devices to acquire IoT data. They also need to store topics for the number of IoT devices in case of topic changing by users. When a user changes topic, the SP needs to know the change. Therefore, SP assigns one Tag ID to multiple topics. The user keeps the correspondence between the assigned Tag ID and the topic in the UFN. Tag ID allows SFN to request multiple data from users in a single subscription.

One of the Tag ID-based pub/sub model issues is that since Tag ID is assigned to multiple topics, it may be disadvantageous for users when a device that publishes private data is associated with a Tag ID.

**3 Table-based Access Control List**

We propose the Table-based ACL to protect private data leakage. Fig. 1 (a) shows
an overview. If \( dev_1 \) is “Deny” to send data to \( SFN_2 \), data transmission is not performed. If it is “Permit”, data transmission is performed.

![Diagram showing the ACL table and data flow]

(a) If data transmission is “Permit” and “Deny”.

(b) IoT Devices to UFN Model.

(c) UFN to SFN Model.

**Fig. 1. Proposed Method.**

In this model, SFNs act as subscribers, and IoT devices act as publishers. The number of SFNs is \( N_{\text{sub}} \), the number of IoT devices is \( N_{\text{pub}} \), and the number of rows in the table is \( N_{\text{table}} \). Then,

\[
N_{\text{table}} = N_{\text{sub}} \times N_{\text{pub}}
\]

(1)

Its computational order is \( O(\log (N_{\text{table}})) \), when the binary search algorithm is used to find the Table-based ACL.

We evaluate the proposed method by simulation to investigate the impact of applying Table-based ACL. In the simulation, we compare the average bandwidth between UFN and SFN and the queuing delay in SFN with/without Table-based ACL. The proposed method is modeled using the queuing theory. Fig. 1 (b) and Fig. 1 (c) shows the model. Here, the number of IoT devices is \( N \). There are one UFN and two SFNs. Publications of IoT data follows the poisson distribution. The arrival rate of the data published from the IoT device is \( \lambda \) [packets/unit time]. The queue in the UFN stores the IoT data. The UFN processes the stored IoT data on a first come first served (FCFS) basis. Next, consider the case of calling IoT data from the UFN to the SFN. The number of installed SFNs is \( K \). \( \mu_{s_i} \) be the average processing rate of the \( i \)-th SFN. If the average arrival rate of the \( i \)-th SFN is \( \lambda_{s_i} \), the relationship between the average processing time of SFN and the load factor is the following equation.

\[
\rho = \frac{\lambda_{s_i}}{\mu_{s_i}} \ (i = 1, 2, ..., K)
\]

(2)
Here, $\rho$ ($0 < 1.0$) indicates the load factor of the fog node. The queue in the SFN stores the IoT data. The average queuing delay $W_{s,i}$ of the $i$-th SFN can also be expressed by the following equation by Little's theorem.

$$W_{s,i} = \frac{1}{\mu_{s,i} - \lambda_{s,i}} + \frac{1}{\mu_{s,i}} \quad \text{[unit time]}$$  \hspace{1cm} (3)

The equations for the average processing rate and queuing delay of UFN are omitted since they are not subject to evaluation. Let the total amount of data received by the SFN be $D_{all}$ and the simulation time be $T_{sim}$. The average bandwidth usage $B_s$ is given by the following equation.

$$B_s = \frac{D_{all}}{T_{sim}} \quad \text{[Mbps]}$$  \hspace{1cm} (4)

4 Evaluation

The simulation parameters are shown in Table I.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulator</td>
<td>NS3 (ns-3.29) [5]</td>
</tr>
<tr>
<td>Model</td>
<td>M/M/1</td>
</tr>
<tr>
<td>Average processing time ($1/\mu_s$)</td>
<td>1, 2, ..., 10 [ms]</td>
</tr>
<tr>
<td>Service discipline</td>
<td>FCFS</td>
</tr>
<tr>
<td>Message Size</td>
<td>100 to 36000 [byte] (Depends on device type)</td>
</tr>
<tr>
<td>Number of Publishers (IoT Devices)</td>
<td>96</td>
</tr>
<tr>
<td>Number of Brokers (UFNs)</td>
<td>1</td>
</tr>
<tr>
<td>Number of Subscribers (SFNs)</td>
<td>2</td>
</tr>
<tr>
<td>Number of Topics</td>
<td>96</td>
</tr>
<tr>
<td>Number of Tag IDs</td>
<td>8</td>
</tr>
<tr>
<td>Simulation Topology</td>
<td>Star topology</td>
</tr>
<tr>
<td>Simulation time</td>
<td>3600 [s]</td>
</tr>
<tr>
<td>Simulation trails</td>
<td>30</td>
</tr>
</tbody>
</table>

$SP_1$ provides the security services and $SP_2$ provides the disaster mitigation service. Each service assigns four Tag IDs to users. The relationship between the user’s IoT device and the Tag ID is shown below.

1) Temperature sensor … **Temp**
2) Humidity sensor … **Humid**
3) Light intensity sensor … **Light**
4) Surveillance camera … **Image**
5) Fire alarm … **Fire**
6) Smoke detector … **Smoke**
7) Thermography … **Thermography**
8) Vibration detector … **Vibration**

Device 1) to 4) are for security services. Device 5) to 8) are for disaster mitigation services. Devices 1),2) 3) and 8) publish data to the UFN at an average interval of 1 second. Devices 4) to 7) publish data to the user fog node at an average interval of 60 seconds. In addition, assuming a disaster, devices in 5) and 6) publish data at an average interval of 1 second from the simulation elapsed time of 2000 [sec]. The message size for devices 1) 2) 3) and 8) is 100 to 300 [Bytes]. Device in 5) and 6) publishes data from 100 to 300 [Bytes], device 4) publishes data from 10000 to 35000 [Bytes], and device 7) publishes data from 15000 [Bytes] to 18000
The Table-based ACL sets “Permit” on devices 1) to 4) for the security service, and it sets “Permit” on devices 5) to 8) for the disaster mitigation service. It sets “Permit” for are all “Permit” randomly on other devices.

The simulation results are shown in Fig. 2.

![Average bandwidth usage.](image1)

![Average queuing delay.](image2)

Fig. 2. Simulation results.

Fig. 2 (a) shows that the average bandwidth usage between SFNs and UFNs of $SP_1$ and $SP_2$. It reduced by about 20% when Table-based ACL is applied. This is because the Table-based ACL suppressed the transmission of private data. Increasing the load factor of the SFN did not affect the average bandwidth usage. Fig. 2 (b) shows the decrease in queuing delay comparing before and after applying Table-based ACL. This is because the application of Table-based ACL has extended the data transmission interval by not transmitting data that has been “Deny”. This is because the queue length has been shortened by not sending data that has been “Deny” due to the application of Table-based ACL.

### 5 Conclusion

In this paper, we propose a Table-based ACL to protect private data leakage, and evaluate the SFN side by simulation. The simulation evaluation shows that the bandwidth usage between UFN and SFN can be reduced. In addition, the queuing delay of SFNs increased exponentially, and when Table-based ACL were applied, the queuing delay decreased up to about 60% compared to that before the application. This simulation is only to show the general trend of the proposed method; the simulation results will vary depending on the number of IoT devices, the number of SFNs, and the number of UFNs.
In the future, we plan to implement the system on a real machine to evaluate it in detail.

**Acknowledgments**

Part of this research was supported by JSPS KAKENHI Grant Number JP18K11276.